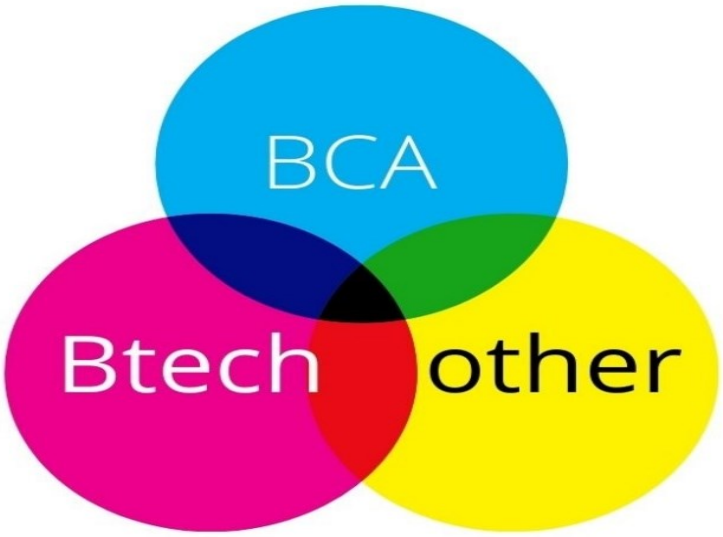
LEARN DATA STRUCTURE IN AN EASY WAY



EASY DATA STRUCTURE USING C LANGUAGE

Ranjot Singh Chahal

Rana Books
Google Play
Pothi.com

Rana Books India

Easy Data Structure Using C Language

RANJOT SINGH CHAHAL

Rana Books India

RANA BOOKS INDIA

Copyright © Ranjot Singh Chahal
All Rights Reserved. 2021 (Rana Books India)

This book has been published with all efforts taken to make the material error-free after the consent of the author. However, the author and the publisher do not assume and hereby disclaim any liability to any party for any loss, damage, or disruption caused by errors or omissions, whether such errors or omissions result from negligence, accident, or any other cause..

While every effort has been made to avoid any mistake or omission, this publication is being sold on the condition and understanding that neither the author nor the publishers or printers would be liable in any manner to any person by reason of any mistake or omission in this publication or for any action taken or omitted to be taken or advice rendered or accepted on the basis of this work. For any defect in printing or binding the publishers will be liable only to replace the defective copy by another copy of this work then available

Rana Books India

Foreword

To begin with, I would like to thank the Almighty for giving me the intellect to create this book.Many authors have contributed to creating this priceless guide. Words are not enough to explain how thankful i am to those who have joined me on this journey.

Ranjot Singh Chahal

Rana Books India

About Book

This book is very easy to read. This book gives a good introduction and complete introduction to data structures and algorithms for beginners. This book is best suited for BCA and BTech readers for the first time, this book covers all data structures subjects of BCA and B.Tech for all computer science students and professionals. Through this book, students will be able to understand the data structure in a very short time. This book has been created after receiving information from many sources and internet

Author: Ranjot Singh Chahal

Rana Books

INDEX

UNIT-1 INTRODUCTION

1.1 BASIC TERMINOLOGY: ELEMENTARY DATA ORGANIZATION 1.1.1 Data and Data Item
- **1.1.2 Data Type**
- **1.1.3 Variable**
- **1.1.4 Record**
- **1.1.5 Program**
- **1.1.6 Entity**
- **1.1.7 Entity Set**
- **1.1.8 Field**
- **1.1.9 File**
- **1.1.10 Key**

1.2 ALGORITHM

1.3 EFFICIENCY OF AN ALGORITHM

1.4 TIME AND SPACE COMPLEXITY

1.5 ASYMPTOTIC NOTATIONS
- **1.5.1 Asymptotic**
- **1.5.2 Asymptotic Notations**
  - 1.5.2.1 Big-Oh Notation (O)
  - 1.5.2.2 Big-Omega Notation (Ω)
  - 1.5.2.3 Big-Theta Notation (Θ)
- **1.5.3 Time Space Trade-off**

1.6 ABSTRACT DATA TYPE

1.7 DATA STRUCTURE
- **1.7.1 Need of data structure**
- **1.7.2 Selecting a data structure**
- **1.7.3 Type of data structure**
  - 1.7.3.1 Static data structure
  - 1.7.3.2 Dynamic data structure
  - 1.7.3.3 Linear Data Structure
  - 1.7.3.4 Non-linear Data Structure

1.8 A BRIEF DESCRIPTION OF DATA STRUCTURES 1.8.1 Array
- **1.8.2 Linked List**
- **1.8.3 Tree**
- **1.8.4 Graph**
- **1.8.5 Queue**
- **1.8.6 Stack**

1.9 DATA STRUCTURES OPERATIONS

1.10 ARRAYS: DEFINITION
- **1.10.1 Representation of One-Dimensional Array**

Infra Books India

1.10.2 Two-Dimensional Arrays
1.10.3 Representation of Three& Four Dimensional Array
1.10.4 Operations on array
1.10.5 Applications of array
1.10.6 Sparse matrix
1.11 STATIC AND DYNAMIC MEMORY ALLOCATION
1.12 LINKED LIST
1.12.1 Representation of Linked list in memory
1.12.2 Types of linked lists
1.12.3 The Operations on the Linear Lists
1.12.4 Comparison of Linked List and Array
1.12.5 Advantages
1.12.6 Operations on Linked List
1.12.7 Doubly Linked Lists
1.12.8 Circular Linked List
1.12.9 Polynomial representation and addition
1.13 GENERALIZED LINKED LIST

UNIT-2 STACKS

2.1 STACKS
2.2 PRIMITIVE STACK OPERATIONS
2.3 ARRAY AND LINKED IMPLEMENTATION OF STACK IN C
2.4 APPLICATION OF THE STACK (ARITHMETIC EXPRESSIONS)
2.5 EVALUATION OF POSTFIX EXPRESSION
2.6 RECURSION
2.6.1 Simulation of Recursion
2.6.1.1Tower of Hanoi Problem
2.6.2 Tail recursion
2.7 QUEUES
2.8 DEQUEUE (OR) DEQUE (DOUBLE ENDED QUEUE)
2.9 PRIORITY QUEUE

UNIT-3 TREES

3.1 TREES TERMINOLOGY
3.2 BINARY TREE
3.3 TRAVERSALS
3.4 BINARY SEARCH TREES
3.4.1 Non-Recursive Traversals
3.4.2 Array Representation of Complete Binary Trees
3.4.3 Heaps
3.5 DYNAMIC REPRESENTATION OF BINARY TREE
3.6 COMPLETE BINARY TREE
3.6.1 Algebraic Expressions
3.6.2 Extended Binary Tree: 2-Trees
3.7 TREE TRAVERSAL ALGORITHMS

Kalpa Books India

3.8 THREADED BINARY TREE
3.9 HUFFMAN CODE

UNIT-4 GRAPHS

4.1 INTRODUCTION
4.2 TERMINOLOGY
4.3 GRAPH REPRESENTATIONS
4.3.1 Sequential representation of graphs
4.3.2 Linked List representation of graphs
4.4 GRAPH TRAVERSAL
4.5 CONNECTED COMPONENT
4.6 SPANNING TREE
4.6.1 Kruskal's Algorithm
4.6.2Prim's Algorithm
4.7 TRANSITIVE CLOSURE AND SHORTEST PATH ALGORITHM 4.6.1 Dijikstra's Algorithm
4.6.2Warshall's Algorithm
4.8 INTRODUCTION TO ACTIVITY NETWORKS

UNIT-5 SEARCHING

5.1 SEARCHING
5.1.1 Linear Search or Sequential Search
5.1.2 Binary Search
5.2 INTRODUCTION TO SORTING
5.3 TYPES OF SORTING
5.3.1 Insertion sort
5.3.2 Selection Sort
5.3.3 Bubble Sort
5.3.4 Quick Sort
5.3.5 Merge Sort
5.3.6 Heap Sort
5.3.7 Radix Sort
5.4 PRACTICAL CONSIDERATION FOR INTERNAL SORTING
5.5 SEARCH TREES
5.5.1 Binary Search Trees
5.5.2 AVL Trees
5.5.3 M-WAY Search Trees
5.5.4 B Trees
5.5.5 B+ Trees 5.6 HASHING
5.6.1 Hash Function
5.6.2 Collision Resolution Techniques
5.7 STORAGE MANGMENT 5.7.1 Garbage Collection 5.7.2Compaction

Rama Books India

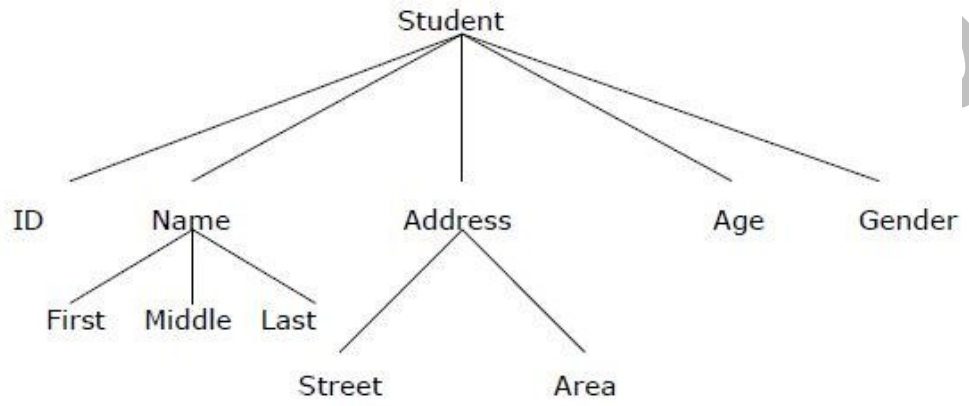
Rana Books India

UNIT-1

1.1 BASIC TERMINOLOGY: ELEMENTARY DATA ORGANIZATION

1.1.1 Data and Data Item

Data are simply collection of facts and figures. Data are values or set of values. A data item refers to a single unit of values. Data items that are divided into sub items are group items; those that are not are called elementary items. For example, a student's name may be divided into threesub items – [first name, middle name and last name] but the ID of a student would normally be treated as a single item. In the above example (ID, Age, Gender, First, Middle, Last, Street, Area) are elementary data items,



whereas (Name, Address) are group data items.

1.1.2 Data Type

Data type is a classification identifying one of various types of data, such as floating-point, integer, or Boolean, that determines the possible values for that type; the operations that can be done on values of that type; and the way values of that type can be stored. It is of two types: Primitive and non-primitive data type. Primitive data type is the basic data type that is provided by the programming language with built-in support. This data type is native to the language andis supported by machine directly while non-primitive data type is derived from primitive data type. For example- array, structure etc.

1.1.3 Variable

It is a symbolic name given to some known or unknown quantity or information, for the purpose of allowing the name to be used independently of the information it represents. A variable name in computer source code is usually associated with a data storage location and thus also its contents and these may change during the course of program execution.

1.1.4 Record

Collection of related data items is known as record. The elements of records are usually called fields or members. Records are distinguished from arrays by the fact that their number of fieldsis typically fixed, each field has a name, and that each field may have a different type.

Rahul Bo

1.1.5 Program

A sequence of instructions that a computer can interpret and execute is termed as program.

1.1.6 Entity

An entity is something that has certain attributes or properties which may be assigned some values. The values themselves may be either numeric or non-numeric.
Example:

Attributes:	Name	Age	Gender	Social Society number
Values:	Hamza	20	M	134-24-5533
	Ali Rizwan	23	M	234-9988775
	Fatima	20	F	345-7766443

1.1.7 Entity Set

An entity set is a group of or set of similar entities. For example, employees of an organization, students of a class etc. Each attribute of an entity set has a range of values, the set of all possible values that could be assigned to the particular attribute. The term *"information"* is sometimes used for data with given attributes, of, in other words meaningful or processed data.

1.1.8 Field

A field is a single elementary unit of information representing an attribute of an entity, a record is the collection of field values of a given entity and a file is the collection of records of the entities in a given entity set.

1.1.9 File

File is a collection of records of the entities in a given entity set. For example, file containing records of students of a particular class.

1.1.10 Key

A key is one or more field(s) in a record that take(s) unique values and can be used to distinguish one record from the others.

1.2 ALGORITHM

A well-defined computational procedure that takes some value, or a set of values, as input and produces some value, or a set of values, as output. It can also be defined as sequence of computational steps that transform the input into the output.

An algorithm can be expressed in three ways:-
(i) in any natural language such as English, called pseudo code.
(ii) in a programming language or
(iii) in the form of a flowchart.

1.3 EFFICIENCY OF AN ALGORITHM

In computer science, algorithmic efficiency are the properties of an algorithm which relate to the amount of resources used by the algorithm. An algorithm must be analyzed to determine its

Rama Books India

resource usage. Algorithmic efficiency can be thought of as analogous to engineering productivity for a repeating or continuous process.

For maximum efficiency we wish to minimize resource usage. However, the various resources (e.g. time, space) can not be compared directly, so which of two algorithms is considered to be more efficient often depends on which measure of efficiency is being considered as the most important, e.g. is the requirement for high speed, or for minimum memory usage, or for some other measure. It can be of various types:

- Worst case efficiency: It is the maximum number of steps that an algorithm can take for any collection of data values.
- Best case efficiency: It is the minimum number of steps that an algorithm can take any collection of data values.
- Average case efficiency: It can be defined as
  - the efficiency averaged on all possible inputs
  - must assume a distribution of the input
  - We normally assume uniform distribution (all keys are equally probable)

If the input has size *n*, efficiency will be a function *of n*

1.4 TIME AND SPACE COMPLEXITY

Complexity of algorithm is a function of size of input of a given problem instance which determines how much running time/memory space is needed by the algorithm in order to run to completion.
Time Complexity: Time complexity of an algorithm is the amount of time it needs in order to run to completion.
Space Complexity: Space Complexity of an algorithm is the amount of space it needs in order to run to completion.
There are two points which we should consider about computer programming:-
(i) An appropriate data structure and
(ii) An appropriate algorithm.

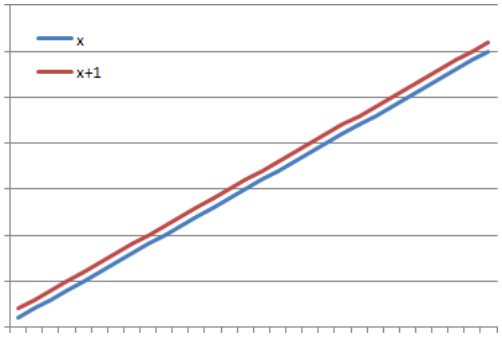
1.5 ASYMPTOTIC NOTATIONS

1.5.1 Asymptotic

It means a line that continually approaches a given curve but does not meet it at any finite distance.

Example

x is asymptotic with x + 1 as shown in graph.



Asymptotic may also be defined as a way to describe the behavior of functions in the limit or without bounds.

Let f(x) and g(x) be functions from the set of real

Rana Books India

numbers to the set of real numbers.

We say that f and g are asymptotic and write f(x) ≈ g(x) if

$\lim_{x \to \infty}$ f(x) / g(x) = *c* (constant)

1.5.2 Asymptotic Notations

1.7.2.1 Big-Oh Notation (O)

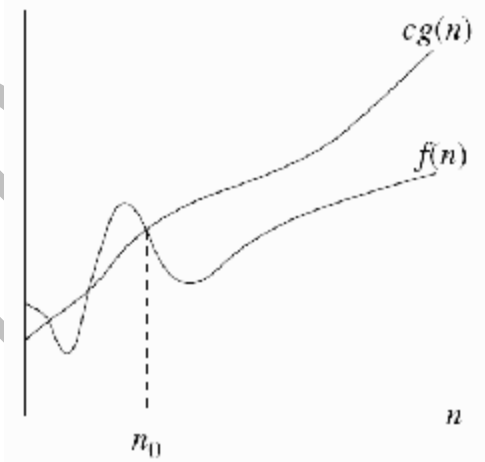
It provides possibly asymptotically tight **upper** bound for f(n) and it does not give best case complexity but can give worst case complexity.

Let f be a nonnegative function. Then we define the three most common asymptotic bounds as follows.

We say that f(n) is Big-O of g(n), written as f(n) = O(g(n)), iff there are positive constants c and n0 such that

$$0 \leq f(n) \leq c\, g(n) \text{ for all } n \geq n0$$

If f(n) = O(g(n)), we say that g(n) is an upper bound on f(n).



Example - $n^2 + 50n = O(n^2)0$

≤ h(n) ≤ *c*g(n)

$0 \leq n^2 + 50n \leq cn^2$

$0/n^2 \leq n^2/n^2 + 50n/n^2 \leq cn^2/n^2$ Divide by n^2

$0 \leq 1 + 50/n \leq c$ Note that $50/n \rightarrow 0$ as $n \rightarrow \infty$

Rana Books India

Pick n = 50

$0 \le 1 + 50/50 = 2 \le c = 2$ With c=2

$0 \le 1 + 50/n_0 \le 2$ Find n_0

$-1 \le 50/n_0 \le 1$

$-20n_0 \le 50 \le n_0 = 50$ $n_0=50$

$0 \le n^2 + 50n \le 2n^2$ $\forall\ n \ge n_0=50$, c=2

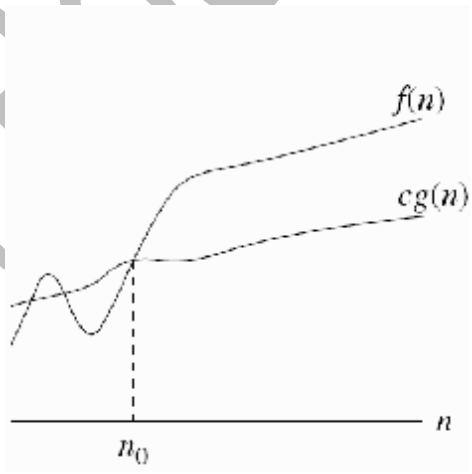
1.7.2.2 Big-Omega Notation (Ω)

It provides possibly asymptotically tight **lower** bound for f(n) and it does not give worst case complexity but can give best case complexity

f(n) is said to be Big-Omega of g(n), written as f(n) = Ω(g(n)), iff there are positive constants c and n0 such that

$$0 \le c\ g(n) \le f(n) \text{ for all } n \ge n0$$

If f(n) = Ω(g(n)), we say that g(n) is a lower bound on f(n).



Example - $n^3 = \Omega(n^2)$ with c=1 and $n_0=1$

$0 \le cg(n) \le h(n)$

$0 \le 1*1^2 = 1 \le 1 = 1^3$

$0 \le cg(n) \le h(n)$

Rana Books India

$0 \leq cn^2 \leq n^3$

$0/n^2 \leq cn^2/n^2 \leq$

$n^3/n^20 \leq c \leq n$

$0 \leq 1 \leq 1$ with c=1 and n_0=1 since n

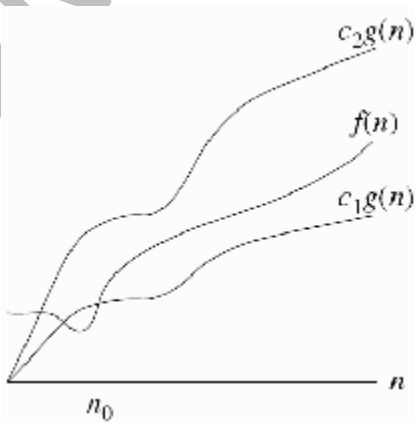
$\lim_{n \to \infty}$ increases.n = ∞

1.7.2.3 Big-Theta Notation (Θ)

• We say that f(n) is Big-Theta of g(n), written as f(n) = Θ(g(n)), iff there are positive constantsc1, c2 and n0 such that

$$0 \leq c1\ g(n) \leq f(n) \leq c2\ g(n) \text{ for all } n \geq n0$$

Equivalently, f(n) = Θ(g(n)) if and only if f(n) = O(g(n)) and f(n) = Ω(g(n)). If f(n) = Θ(g(n)), we say that g(n) is a tight bound on f(n).



Example - $n^2/2\text{-}2n = \Theta(n^2)$

$c_1g(n) \leq h(n) \leq$

$c_2g(n) c_1n^2 \leq n^2/2\text{-}$

$2n \leq c_2n^2$

$c_1n^2/n^2 \leq n^2/2n^2\text{-}2n/n^2 \leq c_2n^2/n^2$ Divide

by $n^2c_1 \leq 1/2\text{-}2/n \leq c_2$

Rana Books India

click here for full book Ebook:

RANA BOOKS INDIA STORE :

https://ranabooks.stores.instamojo.com/product/284621/easy-data-structure-using-c-language-book-fo

PAYTM EBOOKS STORE :

https://paytmebooks.stores.instamojo.com/product/284666/easy-data-structure-using-c-language-a-compl

Rana Books India